REMARKS

Figure 1 has been objected to because, on page 8, lines 11-13 of the specification, Figure 1 is described as an exemplary embedded system which may be used to practice the method of Figure 3. Further, the Office Action states that this permissive language suggests that this embedded system does not require the method to function and exists separately from it. Further still, the Office Action states that the circuitry shown in Figure 1 is well known.

Claim 16 has been objected to because a typographical error in line 2 results in the phrase "copy the", which should be "copy of the".

The specification has been objected to because the Abstract does not comply with proper language and format requirements.

Claims 17 and 30 have been rejected under 35 U.S.C. §112 as being indefinite for failing to particularly point out and distinctly claim the subject matter, which applicant regards as the invention.

Claims 1, 6, 8-11, 13, 16, 18-21, 31, 32, 37, and 39-42 have been rejected under 35 U.S.C. §102(a) as being anticipated by the "Description of the Prior Art" ("DPA").

Claims 2, 3, 28-30, 33, 34, and 44 have been rejected under 35 U.S.C. §103(a) as being unpatentable over DPA, in view of "Linkers and Loaders" by Levine ("Levine").

Claims 4, 14, and 35 have been rejected under 35 U.S.C. §103(a) as being unpatentable over DPA.

Claims 5, 15, 23, and 36 have been rejected under 35 U.S.C. §103(a) as being unpatentable over DPA, in view of "Understanding Computer: Input/Output" by Time-Life Books ("Time-Life").

Page 16 of 32

Claims 7 and 38 have been rejected under 35 U.S.C. §103(a) as being unpatentable over DPA, in view of Levine, and further in view of U.S. Patent No. 4,974,191 to Amirghodsi et al. ("Amirghodsi").

Claim 17 has been rejected under 35 U.S.C. §103(a) as being unpatentable over DPA, in view of Amirghodsi.

Claim 24 has been rejected under 35 U.S.C. §103(a) as being unpatentable over DPA, in view of Time-Life, and further in view of "Microsoft Computer Dictionary" published by Microsoft Press ("MCD").

Claims 12 and 43 have been rejected under 35 U.S.C. §103(a) as being unpatentable over DPA, in view of U.S. Patent No. 6,237,091 to Firooz et al. ("Firooz").

Claims 22 and 25-27 have been rejected under 35 U.S.C. §103(a) as being unpatentable over DPA, in view of Firooz, and further in view of U.S. Patent No. 5,132,716 to Samuels et al. ("Samuels").

Claims 1-44 remain pending.

Objections to the Figures

With respect to the objection to the drawings, Applicant respectfully traverses.

The Office Action states that Figure 1 is described as an exemplary embedded system which may be used to practice the method of Figure 3. Further, the Office Action states that this permissive language suggests that this embedded system does not require the method to function and exists separately from it. Further still, the Office Action states that the circuitry shown in Figure 1 is well known.

G:\IBM\1545\\14942\amend\14942.am1

The specification on page 8, line 11 has been amended to clearly point out that the embedded system shown in Figure 1 is used to practice the method of Figure 3, <u>in one particular embodiment</u>. Further, a new use for an existing system is patentable. Support for this amendment can be found, at least, on page 10, lines 13-29.

Applicant submits that this rejection is now overcome.

Objection to Claim 16

Claim 16 has been amended to correct the typographical error and properly read "copy <u>of</u> the", as suggested by the Examiner.

Applicant submits that this objection is now overcome.

Objections to the Abstract

The Office Action states that the specification has been objected to because the Abstract does not comply with proper language and format requirements. The Abstract has been amended to reduce the word count to between 50 and 150 words, and to remove legal phraseology.

Applicant submits that this objection has now been overcome.

Rejection of Claims 17 and 30 under 35 U.S.C. §112

The Office Action states that claim 17 recites the limitations "further test" and "the offset" in line 2, and that there is insufficient antecedent basis for these limitations in the claim.

Claim 17 has been amended to provide the proper antecedent basis for the recited limitations.

G:\IBM\1545\14942\amend\14942.am1

The Office Action states that claim 30 recites the limitation "the offsets" in line 1, and that there is insufficient antecedent basis for this limitation in the claim.

Claim 30 has been amended to provide the proper antecedent basis for the recited limitation.

Applicant submits that this rejection is now overcome.

Rejection of Claims 1, 6, 8-11, 13, 16, 18-21, 31, 32, 37, and 39-42 under 35 U.S.C. §102(a)

The Office Action states that the DPA teaches the steps of independent claim 1 where step (a) is disclosed by the text "Thus, the system firmware on the PROM may be updated because the current system firmware is executed from the RAM; step (b) is disclosed by the text "Utilizing code update routines from the firmware update itself has associated advantages"; and step (c) is disclosed by the text "...a code update to the firmware may occur while the system operates normally utilizing one or more other process threads of the firmware, thereby accomplishing a background code update to the firmware." Further, the Office Action states that in the context of normally utilizing threads with a background code update, a process of normally switching tasks in order to provide the background update is described. Since task switching is normally handled by the current code image, normal utilization of threads in terms of task switching is handled by the current code image.

It is respectfully submitted that the cited text from the DPA has been taken out of context and mischaracterized. Cited references must be in interpreted in light of the totality of what they disclose. Specifically, the text "Therefore, by executing firmware from a separate memory device, a code update to the firmware may occur while the system operates normally utilizing one or more other process threads of the firmware, thereby accomplishing a background code

Page 19 of 32

update to the firmware" refers to executing a current code image from a separate memory device, such as a RAM, or PROM, for example, while an incoming code image is received and stored on the only PROM (in the case where the current code image is executing from RAM) or stored on a second PROM (in the case were the current code image is executing from a first PROM). While the incoming code image is being stored, the current code image is free to utilize a process thread or switch between multiple process threads. However, as stated in the DPA (see page 4, lines 14 to page 5, line 29), there exists a problem with executing firmware update routines out of the firmware update. As the firmware update routines (from the incoming code image) are executing, the routines may attempt to call a task switching routine outside the scope of the update routines. This cannot be allowed to happen, as unpredictable results are likely to occur. The cited text from the DPA does not address how to solve this problem, but only states the limitations in the prior art method.

A solution to the above-described problem is taught in Applicant's independent claim 1 now amended. Further, Applicant's independent claim 1 has been amended to clearly point out the claimed solution. Support for the claim amendment can be found, at least, on page 14, lines 1-26 of the specification. Implicit in this description of the invention is the requirement that task switching only be initiated by the current code image and that any task switching functions in the incoming code image update routines be kept from executing. In other words, the function of the update routines of the incoming code image is restricted to updating the firmware. If at anytime a task switching function is required, the update routines of incoming code image relinquish control to the current code image to execute task switching functions.

Specifically, Applicant's independent claim 1 step (c) teaches executing a task switching function, that originates only from the current code image, in order to switch microprocessor

G:\IBM\1545\14942\amend\14942.am1

control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing. The cited text from the DPA only refers to the current code image switching processes while the firmware code is updated by the incoming code image. The cited text does not discuss how to solve the problem caused by update routines from an incoming code image executing task switching functions as part of the update process.

In view of the foregoing, it is respectfully submitted that DPA does not teach or suggest the subject matter recited in Applicant's independent claim 1. Specifically, the DPA does not teach or suggest a method that involves executing a task switching function, that originates only from the current code image, in order to switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing.

Claims 6, and 8-11, which depend directly or indirectly from the independent claim 1, incorporate all of the limitations of the independent claim 1 and are therefore patentably distinct over the DPA for at least those reasons provided for independent claim 1.

Independent claims 13, 31, and 32 have also been amended to clearly point out the claimed invention and recite limitations similar to those recited in independent claim 1, and are therefore patentably distinct over the DPA for at least those reasons provided for independent claim 1.

Claims 16, 18, 21, 37, and 39-42, which depend directly or indirectly from the independent claims 13, 31, and 32, incorporate all of the limitations of the corresponding independent claim and are therefore patentably distinct over the DPA for at least those reasons provided for claims 13, 31, and 32.

Rejection of Claims 2, 3, 28-30, 33, 34, and 44 under 35 U.S.C. §103(a)

The Office Action states that in an analogous environment, Levine teaches that the location of routines within a code segment can be represented by an offset from the beginning of the segment, and that it would have been obvious to use Levine's teaching of relocation offsets with the routines of DPA.

As previously discussed, the DPA does not teach or suggest the subject matter recited in Applicant's independent claims 1, 13, 31, and 32. Specifically, the DPA does not teach or suggest a method or system that involves executing a task switching function, that originates only from the current code image, in order to switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing.

Further, because the DPA does not teach or suggest the subject matter recited in independent claims 1, 13, 31, and 32, and because Levine does not teach or suggest the elements of claims 1, 13, 31, and 32 that the DPA is missing, Levine is irrelevant.

In view of the foregoing, it is respectfully submitted that DPA and Levine, whether taken alone or in combination, do not teach or suggest the subject matter recited in claims 1, 13, 31,

G:\IBM\1545\14942\amend\14942.am1

and 32 as each of these references fails at least to teach or suggest a method or system that involves executing a task switching function, that originates only from the current code image, in order to switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing.

Claims 2, 3, 28-30, 33, and 34, which depend directly or indirectly from the independent claims 1, 13, and 32, incorporate all of the limitations of the corresponding independent claim and are therefore patentably distinct over the DPA in view of Levine for at least those reasons provided for claims 1, 13, and 32.

Independent claim 44 recites limitations similar to those recited in independent claims 1, 13, 31, and 32, and is therefore patentably distinct over the DPA in view of Levine for at least those reasons provided for independent claims 1, 13, 31, and 32.

Rejection of Claims 4, 14, and 35 under 35 U.S.C. §103(a)

As previously discussed, the DPA does not teach or suggest the subject matter recited in Applicant's independent claims 1, 13, and 32. Specifically, the DPA does not teach or suggest a method or system that involves executing a task switching function, that originates only from the current code image, in order to switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing.

G:\IBM\1545\14942\amend\14942.am1

Claims 4, 14, and 35, which depend directly or indirectly from the independent claims 1, 13, and 32, incorporate all of the limitations of the corresponding independent claim and are therefore patentably distinct over the DPA for at least those reasons provided for claims 1, 13, and 32.

Rejection of Claims 5, 15, 23, and 36 under 35 U.S.C. §103(a)

The Office Action states that in an analogous environment, Time-Life teaches that computers gather and distribute digital information using an input/output interface. Further, the Office Action states that Time-life teaches computer systems including a bus, microprocessor, RAM, programmable memory, and I/O interface.

As previously discussed, the DPA does not teach or suggest the subject matter recited in Applicant's independent claims 1, 13, and 32. Specifically, the DPA does not teach or suggest a method or system that involves executing a task switching function, that originates only from the current code image, in order to switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing.

Further, because the DPA does not teach or suggest the subject matter recited in independent claims 1, 13, and 32, and because Time-Life does not teach or suggest the elements of claims 1, 13, and 32 that the DPA is missing, Time-Life is irrelevant.

In view of the foregoing, it is respectfully submitted that DPA and Time-Life, whether taken alone or in combination, do not teach or suggest the subject matter recited in claims 1, 13,

and 32 as each of these references fails at least to teach or suggest a method or system that involves executing a task switching function, that originates only from the current code image, in order to switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and <u>thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing</u>.

Claims 5, 15, 23, and 36, which depend directly or indirectly from the independent claims 1, 13, and 32, incorporate all of the limitations of the corresponding independent claim and are therefore patentably distinct over the DPA in view of Time-Life for at least those reasons provided for claims 1, 13, and 32.

Rejection of Claims 7 and 38 under 35 U.S.C. §103(a)

The Office Action states that in an analogous environment, Amirghodsi teaches testing a pointer for a NULL value to determine validity.

As previously discussed, the DPA does not teach or suggest the subject matter recited in Applicant's independent claims 1 and 32. Specifically, the DPA does not teach or suggest a method or system that involves executing a task switching function, that originates only from the current code image, in order to switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and <u>thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing</u>.

Further, as previously discussed, because the DPA does not teach or suggest the subject

matter recited in independent claims 1 and 32, and because Levine does not teach or suggest the

elements of claims 1 and 32 that the DPA is missing, Levine is irrelevant.

Further still, because the DPA and Levine, whether taken alone or in combination, do not

teach or suggest the subject matter recited in independent claims 1 and 32, and because

Amirghodsi does not teach or suggest the elements of claims 1 and 32 that the DPA and Levine

are missing, Amirghodsi is irrelevant.

In view of the foregoing, it is respectfully submitted that DPA, Levine, and Amirghodsi,

whether taken alone or in combination, do not teach or suggest the subject matter recited in

claims 1 and 32 as each of these references fails at least to teach or suggest a method or system

that involves executing a task switching function, that originates only from the current code

image, in order to switch microprocessor control from executing one or more code update

routines that originate from the incoming code image, to execute a function that originates from

the current code image and thereby prevent a task switching function that originates from the one

or more code update routines that originate from the incoming code image from executing.

Claims 7 and 38, which depend directly or indirectly from the independent claims 1 and

32, incorporate all of the limitations of the corresponding independent claim and are therefore

patentably distinct over the DPA in view of Levine, and further in view of Amirghodsi, for at

least those reasons provided for claims 1 and 32.

Rejection of Claim 17 under 35 U.S.C. §103(a)

As previously discussed, the DPA does not teach or suggest the subject matter recited in

Applicant's independent claim 13. Specifically, the DPA does not teach or suggest a method or

Page 26 of 32

system that involves executing a task switching function, that originates only from the current code image, in order to switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing.

Further, because the DPA does not teach or suggest the subject matter recited in independent claim 13, and because Amirghodsi does not teach or suggest the elements of claim 13 that the DPA is missing, Amirghodsi is irrelevant.

In view of the foregoing, it is respectfully submitted that DPA and Amirghodsi, whether taken alone or in combination, do not teach or suggest the subject matter recited in claim 13 as each of these references fails at least to teach or suggest a method or system that involves executing a task switching function, that originates only from the current code image, in order to switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing.

Claim 17, which depends directly or indirectly from the independent claim 13, incorporates all of the limitations of the independent claim 13 and is therefore patentably distinct over the DPA in view of Amirghodsi for at least those reasons provided for claim 13.

Rejection of Claim 24 under 35 U.S.C. §103(a)

The Office Action states that in an analogous environment, MCD further teaches that the process of integration combines multiple circuit elements on a single chip.

Page 27 of 32

As previously discussed, the DPA does not teach or suggest the subject matter recited in Applicant's independent claim 13. Specifically, the DPA does not teach or suggest a method or system that involves executing a task switching function, that originates only from the current code image, in order to switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing.

Further, as previously discussed, because the DPA does not teach or suggest the subject matter recited in independent claim 13, and because Time-Life does not teach or suggest the elements of claim 13 that the DPA is missing, Time-Life is irrelevant.

Further still, because the DPA and Time-Life, whether taken alone or in combination, do not teach or suggest the subject matter recited in independent claim 13, and because MCD does not teach or suggest the elements of claim 13 that the DPA and Time-Life are missing, MCD is irrelevant.

In view of the foregoing, it is respectfully submitted that DPA, Time-Life, and MCD, whether taken alone or in combination, do not teach or suggest the subject matter recited in claim 13 as each of these references fails at least to teach or suggest a method or system that involves executing a task switching function, that originates only from the current code image, in order to switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing.

Page 28 of 32

G:\IBM\1545\14942\amend\14942.am1

PAGE 33/37 * RCVD AT 7/8/2005 1:23:05 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:5167424366 * DURATION (mm-ss):12-04

Claim 24, which depends directly or indirectly from the independent claim 13,

incorporates all of the limitations of the independent claim 13 and is therefore patentably distinct

over the DPA in view of Time-Life, and further in view of MCD, for at least those reasons

provided for claim 13.

Rejection of Claims 12 and 43 under 35 U.S.C. §103(a)

The Office Action states that in an analogous environment, Firooz teaches resetting a

computer system upon completion of a code update.

As previously discussed, the DPA does not teach or suggest the subject matter recited in

Applicant's independent claims 1 and 32. Specifically, the DPA does not teach or suggest a

method or system that involves executing a task switching function, that originates only from the

current code image, in order to switch microprocessor control from executing one or more code

update routines that originate from the incoming code image, to execute a function that

originates from the current code image and thereby prevent a task switching function that

originates from the one or more code update routines that originate from the incoming code

image from executing.

Further, because the DPA does not teach or suggest the subject matter recited in

independent claims 1 and 32, and because Firooz does not teach or suggest the elements of

claims 1 and 32 that the DPA is missing, Firooz is irrelevant.

In view of the foregoing, it is respectfully submitted that DPA and Firooz, whether taken

alone or in combination, do not teach or suggest the subject matter recited in claims 1 and 32 as

each of these references fails at least to teach or suggest a method or system that involves

executing a task switching function, that originates only from the current code image, in order to

Page 29 of 32

G:\IBM\1545\14942\amend\14942.am1

switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing.

Claims 12 and 43, which depend directly or indirectly from the independent claims 1 and 32, incorporate all of the limitations of the corresponding independent claim and are therefore patentably distinct over the DPA in view of Firooz for at least those reasons provided for claims 1 and 32.

Rejection of Claims 22 and 25-27 under 35 U.S.C. §103(a)

The Office Action states that in an analogous environment, Samuels teaches bootloaders for instructing a processor to execute a code image.

As previously discussed, the DPA does not teach or suggest the subject matter recited in Applicant's independent claim 13. Specifically, the DPA does not teach or suggest a method or system that involves executing a task switching function, that originates only from the current code image, in order to switch microprocessor control from executing one or more code update routines that originate from the incoming code image, to execute a function that originates from the current code image and thereby prevent a task switching function that originates from the one or more code update routines that originate from the incoming code image from executing.

Further, as previously stated, because the DPA does not teach or suggest the subject matter recited in independent claim 13, and because Firooz does not teach or suggest the elements of claim 13 that the DPA is missing, Firooz is irrelevant.

G:\IBM\1545\\14942\amend\14942.am1

Further still, because the DPA and Firooz, whether taken alone or in combination, do not

teach or suggest the subject matter recited in independent claim 13, and because Samuels does

not teach or suggest the elements of claim 13 that the DPA and Firooz are missing, Samuels is

irrelevant.

In view of the foregoing, it is respectfully submitted that DPA, Firooz, and Samuels,

whether taken alone or in combination, do not teach or suggest the subject matter recited in claim

13 as each of these references fails at least to teach or suggest a method or system that involves

executing a task switching function, that originates only from the current code image, in order to

switch microprocessor control from executing one or more code update routines that originate

from the incoming code image, to execute a function that originates from the current code image

and thereby prevent a task switching function that originates from the one or more code update

routines that originate from the incoming code image from executing.

Claims 22 and 25-27, which depend directly or indirectly from the independent claim 13,

incorporate all of the limitations of the independent claim 13 and are therefore patentably distinct

over the DPA in view of Firooz, and further in view of Samuel, for at least those reasons

provided for claim 13.

Page 31 of 32

Conclusion

In view of the foregoing, applicants respectfully requests reconsideration, withdrawal of all rejections, and allowance of all pending claims in due course.

Respectfully submitted,

Steven Fischman
Registration No. 34,594

SCULLY, SCOTT, MURPHY & PRESSER
400 Garden City Plaza
Suite 300
Garden City, New York 11530
(516) 742-4343

SF:BMM:ej

G:\IBM\1545\14942\amend\14942.am1